

前言

微軟在其網站說明：

“商務使用者最多花費 80% 的時間進行資料準備，這會延遲分析和決策的工作。”

Excel 是資料分析的軟體，普遍性高並容易上手，而且舉凡工作中關於數字方面的處理，Excel 都能勝任，所以，深受一般上班族的喜愛。

在資料分析之前，你拿到其他軟體的原始資料格式可能跟 Excel 不同，此時你必須清洗這些資料，以便分析、制表或繪圖所需。另外有些資料格式需要經過加工處理，以便將資料轉換、串接、合併、拆解、上色…，然後進行計算或提醒。但就像 MS 的說明一樣，通常上班族因為不懂如何快速處理資料，所以，造成最重要的分析和決策的工作延宕。

我曾寫了一本書是進階 EXCEL，書名是《Excel 彙總與參照函數精解》，重點在於如何進階應用常見的函數，說明重要函數的操作原理與應用時機。也曾出版《活用 EXCEL 陣列函數》的教學影片，這也是進階函數的應用，重點在講解 EXCEL 陣列公式的操作，讓你能快速與簡便使用進階陣列函數。

而這本書是從簡單到進階整理混亂的資料，讓它們成為有效用並且可以進行分析與決策的資料。作者將網友常提出的問題點分成 6 大部分，包含：

- 文字整理
- 表格整理
- 拆解整理
- 格式整理
- 時間整理
- Power Query 應用

文字整理

儲存格的字串常常不符所需，所以，需要合併或轉換以取得適當的字串。TEXT 是很強大的文字函數，可以進行數值轉換、格式變動、添加特定文字、串接資料、時

間轉換與邏輯判斷等。這一單元將說明強大的 TEXT 函數的功用以及其他函數對文字的處理。

拆解整理

資料拆解、拆字或字串合併是一個大問題，原始資料很多字會連結在一起，必須要擷取適當且需要的字串。通常我們會使用文字函數進行拆字或用資料剖析，但這些方法都有其侷限性。有個網路函數特別有用——FILTERXML，它本來是進行網頁資料的解讀，但我們將它使用在字串擷取方面。所以，這一單元將應用 FILTERXML 的 XPATH 定義來進行字串的擷取，而且探討擷取之後加工計算；當然，還有其他函數的拆解應用。

時間整理

因為 EXCEL 的儲存格可以輸入任何格式類型，自由度很高，但也因此造成時間標示的問題。時間彙總、參照、轉換、格式化在 EXCEL 的應用是一個重要的議題，當然，EXCEL 有時間與日期函數可以處理，但 TEXT 處理時間問題也是一等一的高手。這一單元將進行日期與時間轉換，還有日期、週別與時間的計算。

表格整理

將表格資料重新整理有時是一大工程，表格轉清單、清單轉表格、資料依照數量重複出現…。我們將利用一般函數來解決這些表格資料重複、轉換、移除..等等問題。當然，Power Query 也是這方面的能手，我們在第 6 單元也會適當說明如何利用 PQ 進行表格整理。這一單元將進行表格資料轉移跟資料比對。

格式整理

儲存格根據條件來改變格式通常使用「條件式格式設定」，用在標示與提醒方面，可以改變原格式的顏色、字體、線條、數值、字型等等的設定。這個單元最主要講解如何使用公式來改變顏色。格式設定的函數用法需要有點想像力，畢竟，它是在設定範圍內從上而下，從左而右逐一掃描儲存格，一一判斷來改變格式，而且要配合絕對與相對位置的設定。大部分使用者會搞混它的處理過程，所以，有可能設定公式後，無法取得適當的結果。這一單元將說明一些重要的基本應用，還有解說常見的公式。

POWER QUERY 應用

清洗或整理資料是方便後續的分析步驟，而前面的單元大都是用函數來解決，其實還有功能區的資料剖析、POWER QUERRY，還有快捷鍵 (SHORTCUT、HOTKEY) 操作等。這單元是 PQ 應用，PQ 不是單一獨立的軟體，而是附加其他軟體，例如：EXCEL、POWER BI...

PQ 功能很強大，尤其是工作表的合併與附加，所以，本單元的重點會放在這裡。通常 PQ 可以進行：

- 單工作表
- 多工作表
- 多檔
- 多檔多工作表
- 混合應用

其實了解多工作表轉換、替代、合併與附加應用時，其他都不是問題。

ETL 是分析資料之前的動作，根據維基百科解釋：

“是英文 Extract-Transform-Load 的縮寫，用來描述將資料從來源端經過擷取 (extract)、轉置 (transform)、載入 (load) 至目的端的過程。”

而微軟認為：

“使用 Power Query，您可以執行擷取、轉換和載入 (ETL) 處理資料。”

這本書重點在利用函數進行從初階到高階的資料清洗與整理，但有時可以使用更簡便或函數無法處理的狀況，所以，加入一些功能區操作、PQ 與快捷鍵的介紹。

如果再配合《Excel 彙總與參照函數精解》這本書的操作，我相信你的 EXCEL 應用將會如虎添翼，協助你快速準確地完成公司的任務。

使用 TEXT 進行資料轉換

TEXT 屬於文字函數，能夠進行數字轉換、時間轉換、邏輯判斷、文數轉換與添加文字。我們將在後面的單元解析這些功能。TEXT 是 Excel 最複雜的函數之一，只要能搞懂它的運作規則，就能大大提昇你的資料清洗與整理功力。

本章重點

- 1.1 數值 - 分數轉換
- 1.2 數值 - 符號字元
- 1.3 數值應用
- 1.4 將阿拉伯數字轉為國字
- 1.5 將國字轉為阿拉伯數字
- 1.6 添加文字
- 1.7 添加文字進階版
- 1.8 一格一數字的表單合計數字轉國字

01 數值 - 分數轉換



數值格式轉換是 TEXT 的強項，本節來檢視分數轉換的狀況。

首先，TEXT 的語法是：

```
TEXT(value, format_text)
```

value：儲存格要剖析的值。

format_text：進行值的格式轉換。

開啟「1.1 TEXT 格式代碼 - 數值 - 分數轉換.xlsx」。

A	B	C	D
2	資料	格式	Text
3	2.2	# ??/?	2 1/5
4	2.333	# ??/?	2 1/3
5	2.7	# ??/??	2 7/10
6	2.4	# ??/?	12/5
7	2.5	# ??/?	5/2
8	2.5	# ?/4	2 2/4
9	5.49	# ??/??	5 25/51
10	3.21	# ?/100	3 21/100
11	0.397	0/5	2/5
12	2.533	# 0/5	2 3/5
13	0.51	0%	51%
14	0.5335123	0.0%	53.4%
15	2 1/2	@	2.5

首先，B 欄是資料，也就是 TEXT 的第 1 引數 value，而 C 欄是轉換 B 欄的格式代碼，也就是 TEXT 的第 2 引數 format_text，format_text 很複雜，功能非常強大，D 欄

是轉換後的答案。在 C3 可以看到 "# ?/?"，這些代碼代表許許多多的涵義。以上面表格中的 C 欄可以看到：

#：占位代碼，如果是 `TEXT(1234.56, "#,###.###")`，返回 1,234.56，最後面的代碼 # 在 value 沒有，所以不會顯示，只到 56 而已。當然你可以用 "#,##"，它返回的是 1,234.6，只顯示小數點後面 1 個位數，而原來有 2 位數，判斷小數點第 2 位數以四捨五入進位，第 2 位數就消除。

0：占位代碼，如果是 `TEXT(1234.56, "00,000.000")`，返回 01,234.560，它跟 # 類似，如果是數值，直接顯示數值；如果沒有的話，就填補 0。

?：占位代碼，它跟上面兩個類似，也會填補空位，可以根據小數點上下兩儲存格對齊，如 `TEXT(1234.56, "?.??")` 與 `TEXT(1234.5, "?.??")`。

/：可以是除號，但只是顯示而已，不是運算符號，也可以成為帶分數或假分數符號，或者是日期分隔符號代碼。

%：百分比代碼。

@：文字代碼。

當然，還有日期代碼、時間代碼、貨幣符號…。我們將在後面章節一一說明。

接下來看 C3 的格式是 # ?/?，#?0 三個都可以使用，只要在 # 與 ? 之間要空一格。使用 / 代碼，2.2 的整數保留，而小數部分會跟 / 後面的代碼數來決定分數的顯示，小數是 0.2，格式 / 後面分母代碼只有 1 個 ?，顯示 2/10，約分之後成為 1/5，整數保留就是 2 1/5。

B4 是 2.333，整數是 2 與小數是 3.33/10，取進位數是 2 1/3。

B6 是 2.4，而格式是 #??，中間沒空格，所以轉成假分數，有空格是帶分數，答案是 12/5。

B8 是 2.5，格式是 # ?/4，分母強制是 4，所以得到帶分數是 2 2/4，並不是 2 1/2。

B9 是 5.49，格式是 # ??/??，分母代碼是 2 個 ?，表示分母是 2 個數， $25/51 \approx 0.4901 \approx 0.49$ 。這裡分子的代碼多寡與不同是有差別，多一個 ? 就會多空一格。如果本來空一格，格式是 ???，就會多空一格，以此類推，而 # 幾個都是一樣空一格

而已。至於 0 的話，超過的地方就會填補 0。如果是 # 000/??，答案是 5 025/51；如果是 # 000/0，答案是 5 001/2，要求分母是一個數字，所以是 $1/2=0.5 \div 0.49$ 。

B13 是 0.51，格式是 0%，0 代碼是原來數值，% 代碼是以百分比顯示，所以答案是 51%。

B14 是 0.533512，格式是 0.0%，要求小數點後面只有一個數字，並且以百分比顯示，小數點後第四位數是 5，四捨五入，所以答案是 53.4%。

B15 是 2 1/2，格式是 @，返回原來的樣貌，所以答案是 2.5。

02 數值 - 符號字元



你可以插入很多符號在字串當中，來表示另一層意義。TEXT 有些內定代碼，在插入符號當中注意代碼的應用，否則會產生不是自己想要的。如前面所提的 0，如果你只是單純想要顯示 0，就必須使用強制符號 ! 或 \，如 !0 或 \0，強制顯示 0。

開啟「1.2 TEXT 格式代碼 - 數值 - 符號字元.xlsx」。

	A	B	C	D
2		資料	格式	Text
3		32	000.00	032.00
4		2.376	-.#00	-2.38
5		55.25	\$#.#	\$55.3
6		543.21	\$#	\$543
7		4567	\$0,0.00	\$4,567.00
8		22.159	0.#	22.2
9		204	!r0c0	r20c4
10		104	\r0c00	r1c04
11		12345678	0.00e+00	1.23e+07
12		12345678	#0.0E+0	12.3E+6
13		1234	¢0,0	¢1,234
14		1234	£0,0	£1,234
15		1234	¥0,0	¥1,234
16		1234	€0,0	€1,234

在前一節曾經說明代碼 0 的功能，就如 D3 所示會填補原數值沒有的空位。B3=32，C3=000.00，就會把 0 補上空位。在儲存格輸入電話號碼，如 0936xxxxxx 或者 028765xxxx，Excel 把它當成一組數字，所以輸入之後 0 會消失，這時可以用 TEXT 或用 Ctrl+1 的數值 → 特殊，來解決這個問題。

-(減號)、\$(錢號)都可以添加以便產生會計符號，如 D4、D5，而,(逗號)是千位符號，所以 C7=\$0,0.00，會得到 D7 的 \$4,57.00 的答案。

C9=!r0c0 的應用，會在「第 3 章 座標法」做詳細的說明，這在取得表格中的字串並重新排列會產生意想不到的效果。r 是 TEXT 內定代碼，表示中華民國曆轉西元曆的前置碼，會在第 6 章的「時間整理」詳細說明。因為 r 是內定代碼，但我們想要顯示 r，所以 r 前面需要有個前置符號! 或 \，強制顯示 r。B9=204，C9=!r0c0，產生 D9 的 r20c4。Excel 的位址樣式有 2 種，一個是 A1；另一個是 R1C1，所以 r20c4 就是列 20 與欄 4，A1 樣式是 D20。接下來就可以透過 INDIRECT 取得位址的值。

D11 是科學記號標記法，是用在比較長的數字將它縮短，便於閱讀，但精確度就比較模糊。B11 是 8 位數，C11=0.00e+00，整數只有 1 位數，小數有 2 個，因為總共是 8 位數，e+00 就轉換成 e+07。

C12 是 #0.0E+0，整數有 2 位數，所以 D12=12.3E+6。

當然，也可以在數值之前加上貨幣符號，C13= ¢ 0,0，D13= ¢ 1,234。你可以用 ALT+162 顯示 ¢ (Cent) 符號，也可以按 Ctrl+1 → 數值 → 貨幣 → 符號選擇適當的貨幣符號。如果是台幣可以用 \$ 或者 NT\$，函數是 =TEXT(12345,"!NT\$0,0")，N 是內定代碼，所以用 ! 的強制符號強制顯示。其他幣別的符號如下表。

日幣與人民幣	¥	ALT+165
歐元	€	Alt-128
美金	US\$	無
英鎊	£	Alt-163
港幣	HK\$	無

03 數值應用



這節要來說明一些 TEXT 數值應用上的小案例，關係第 2 引數 `format_text` 中的其他函數與陣列函數配合。當然，以後章節會有更進階的應用，將 TEXT 與其他函數混合應用發揮更強大的問題解決能力。

開啟「1.3 TEXT 格式代碼 - 數值 - 小案例.xlsx」。

A	B	C	D
2	資料	格式	Text
3	1	00-00	01-02
4	2	00-00	02-13
5	13	00-00	13-05
6	5		
7	2.3567	{0,1}%^5	0.3567
8			0.3567
9	1	!No!.&REPT(0,4)	No.0001
10	2		No.0002
11	13		No.0013
12	132	!r00c0	456
13	456	REPT("~?w",3)	£4 £5 £6
14	23x105x52	REPT("000x",2)&"000"	023x105x052

D1:D5 是日期格式，上節曾說過用代碼來改變 value 的格式，加上 -（橫槓、負號或減號）可以轉為日期格式。D3=TEXT(B3*100+B4,C3)，B3*100+B4=102，而 `format_text` 是 00-00，橫槓後面是兩個 0，它會根據橫槓後面來分割 value，因此 102 先切割 02，再把其餘的值排放在橫槓前面，答案就是 01-02。如果 `format_text=0-0`，則是 10-2。它會將此種格式視為日期格式，所以，MONTH(D3)=1，取得 D3 的月份。但 MONTH(D5)=5，而 D5 的月份在橫槓的後面，因為沒有 13 月份，所以，Excel 自動判斷日期格式的月份是在後面。當然，如果是 13-15 都不是日期格式，就會被判別為一般文字格式，就無法取得月份。還有沒有年度就會判別今年度。

format_text 也可以用陣列形式，B7=2.3567，我們要取得小數部分 0.3567。format_text 是 {0,1}%^5，{0,1} 是常數陣列 (逗號形成橫列，而 ; 分號形成直欄)，TEXT(B7,{0,1}) 得到答案會是 2 跟 1 兩個答案，單一個 0 沒有小數點會取得整數部分，而其他數字就是其他數字，所以，1 就是 1，3 就是 3。{0,1}% 是 {0,0.01}，取得答案是 2 跟 2.41，小數後面兩個數字，B7 是小數點後四個數字，所以，我們使用 %^5 來增加小數點後的位數，形成 {0,0.0000000001}，產生答案是 {"2","2.3567000001"}，取得小數需要大數值扣掉小數值，所以，要乘上 {-1,1}，產生 {-2,2.3567000001}，最後用 SUM 加總就是 0.3567，最後一個 1 利用減少小數位數去除即可。當然，^5 也可改為 ^3 就足夠滿足位數的需求。

除了可用常數陣列以外，也可以用函數。B9:B11 是序號，要轉換成 D9:D11 的型態，所以，format_text 是 !No!.&REPT(0,4)，因為 N 與 . 是內定代碼，要用 ! 強制顯示 N 與 .。REPT 是文字重複次數，語法是：

```
REPT(text, number_times)
```

所以是重複 0 四次，產生 "!No!.0000"，得到答案就是 No.0001。

C12 的格式是 !r00c0，前一節曾經說明這是 R1C1 樣式，透過 INDIRECT 取得該位址的值。B12=123，而 D12 是 256，TEXT(B12,C12) 是 "r13c2"，就是 B13，B13 是 456，所以，INDIRECT("r13c2") 就是反映 B13 的值，答案是 456。INDIRECT 的語法是：

```
INDIRECT(ref_text, [a1])
```

ref_text 是參照文字部分。

[a1] 是 0 的話，R1C1 樣式，可以省略 0，但保留逗號；而 1 的話，是 A1 樣式，可以省略。

B13 是 456，期望達到 D13 的 £4 £5 £6。format_text 是 REPT("~?w",3)，所以 TEXT 會得到 ~4w~5w~6w，~w 是符號，一樣顯示 ~w。因為是重複三次，所以，? 代碼就會代表 456 個別的值。我們要將 ~ 換成 £ 英鎊符號，將 w 換成空格，而符號 ~w 以任何不是內定的代碼都可以，畢竟我們要用函數將他們轉換，SUBSTITUTE 語法是：

```
SUBSTITUTE(text, old_text, new_text, [instance_num])
```

text 是文字型字串。

old_text 是文字中那些字串需要替代。

new_text 是替代成新字串。

[instance_num] 是指定第幾個字元替代，可省略。

D13 公式是：

```
SUBSTITUTE (③
  SUBSTITUTE (②
    TEXT (B13, REPT ("~?w", 3)), ①
    "~",
    "£"
  ),
  "w",
  " "
)
```

1. 將字串重複三次，如上所述。
2. 將 £ 代替 ~。
3. 將空格代替 w。

如果中間不要用空格區隔，以逗號 (,) 區隔，就必須考慮逗號 (,) 的位置。公式是：

```
MID (SUBSTITUTE (TEXT (B13, REPT ("~?", 3)), "~", "£"), 2, 10)
```

MID 是取得中間的字串，語法是：

```
MID(text, start_num, num_chars)
```

text 是文字型字串。

start_num 是從第幾個字元開始。

num_chars 是取得幾個字元。

這裡 `format_text` 只用 `REPT("~?",3)`，我們只替代一個就夠了，因為 `SUBSTITUTE` 所返回的值是 ", £4, £5, £6"，逗點 (,) 放在前面，所以，我們可以從第 2 個字元開始，就會排除第 1 個字元 (,)，至於 `num_chars=10`，超過後面字元數，仍以最後幾個字元數為準，所以，`num_chars` 可以設定大一點的數字，字串長度就可以彈性反應。如果逗號 (,) 在後面，" £4, £5, £6,"，就比較麻煩，因為字串長短不一，所以，先要用 `LEN` 判斷字元數，再減掉 1，然後，用 `LEFT` 得到答案，這樣操作的公式會很長。

`B14=23x105x52` 是才積的立方長寬高的樣式，想要成為 `D14` 的 3 位數的樣式，公式是：

```
TEXT (④
    SUM (③
        MID (②
            SUBSTITUTE (①
                B14,
                "x",
                REPT(" ",10)
            ),
            {1,12,23},
            10
        ) * 10^{6,3,0}
    ),
    REPT("000\x",2) & "000"
)
```

1. 將 `x` 替換成 10 個空格。所以，每組數值 (23、105 與 52) 中間有 10 個空格。
2. 透過 `MID` 擷取 "23 105 52" 中的數值部分，開始從第 1、12 與 23 個字元擷取，共取 10 個。`*10^{6,3,0}` 的意思是，每組數值以 1 後面 6 個 0、3 個 0 與 0 個 0 相乘。返回 {23000000,105000,52}。
3. 將 {23000000,105000,52} 加總，得到 23105052。
4. `TEXT` 的 `format_text` 是 `REPT("000\x",2)&"000"`，這個答案是 "000\x000\x000"，`\` 是強制顯示 `x`，3 個 0 一組，得到 023x105x052。

04 將阿拉伯數字轉為國字



由於文件需求關係，常需要將阿拉伯數字轉為國字，而國字的數字也有兩種形式，如一二三與壹貳參。這節試圖用 TEXT 的 format_text 設定解決這個問題點。

開啟「1.4 TEXT 格式代碼 - 文數轉換.xlsx」。

A	B	C	D
2	資料	格式	Text
3	12345	[DBNum1]	一萬二千三百四十五
4	12350	[DBNum2]	壹萬貳仟參佰伍拾
5	12345	[DBNum3]	1 萬 2 千 3 百 4 十 5
6	12345	[DBNum1]#	一二三四五
7	12345	[DBNum2]0	壹貳參肆伍
8	12345	[DBNum3]?0	1 2 3 4 5.0
9	251200	0萬	25萬1,200元
10	1234	0萬	1,234元
11	1234.5	[DBNum1]	一千二百三十四點五
12	1235.5	[DBNum2]	壹仟貳佰參拾伍點伍
13	1236.5	[DBNum3]	1 千 2 百 3 十 6 點 5
14	1237.5	[DBNum4]	1237點5
15	1238.5	[DBNum1]0.0	一二三八點五
16	1239.5	[DBNum2]0!0	壹貳肆點零
17	1240.5	[DBNum3]0.0	1 2 4 0 點 5
18	1241.5	[DBNum4]0.0	1241點5

通常格式應用 [DBNum] 來進行阿拉伯數字轉為國字。

[DBNum1] 是轉為一二三。

[DBNum2] 是轉為壹貳參。

[DBNum3] 是全形的 1 2 3。

上面 3 種型態都會加入十、百、千、萬…位數。如 D3:D5。

D6 的格式是 [DBNum1]#，後面加上 #(? 與 0 均可) 能去掉十百千萬…的國字。

D8 是用 ?.0 就可以加入小數部分。

如果一樣是阿拉伯數字，但以萬為分隔點就需要應用函數功能。

LEFT 是根據指定數字取得左邊的字元，語法是：

```
LEFT(text, [num_chars])
```

text 是文字字串。

[num_chars] 是從左邊算起擷取幾個字元。

RIGHT 是根據指定數字取得右邊的字元，語法是：

```
RIGHT(text, [num_chars])
```

text 是文字字串。

[num_chars] 是從右邊算起擷取幾個字元。

D9 的公式是：

```
TEXT(②
    LEFT(B9, LEN(B9) - 4), ①
    C9
)
&
TEXT(④
    RIGHT(B9, 4), ③
    "0,0元"
)
```

首先，要取得萬位以上合併千位以下，所以透過 LEFT 與 RIGHT 分別擷取。

1. 數值長度不固定，所以，先要用 LEN 判斷幾個字元，然後扣掉 4，4 是千位以下，接下來使用 LEFT 擷取 C9=251200 左邊算起 2 個字元 (6-4)，答案是 25。
2. 接下來，TEXT 的 format_text 是 0 萬，所以，答案就是 25 萬。

3. RIGHT 取得千位以下，就是 4 個字元，所以，答案是 1200。
4. 然後，TEXT 的 format_text 是 0,0 元，所以，答案是 1,200 元。最後合併 2 的 TEXT。

如果有小數點的話，不能在用點 (.) 代碼，必須將點 (.) 轉換國字。

D11 是：

```
SUBSTITUTE(TEXT(B11,C11),".","點")
```

用 TEXT 轉換之後，只要透過 SUBSTITUTE 將「.」轉為「點」即可。

另外，除了用 TEXT 可以轉換之外，也可以用隱藏的函數 NUMBERSTRING。如 NUMBERSTRING(12345,1) 得到的答案跟 [DBNum1] 一樣。

05 將國字轉為阿拉伯數字



有些行業會有文數字轉移的需求，包含反向轉換（一二三轉 123），如何在票據列印中文數字等。我們簡化這些問題，直接用 TEXT 來處理。

開啟「1.5 TEXT 格式代碼 - 文數轉換 - 小案例.xlsx」。

A	B	C	D
2	資料	格式	Text
3	3456	[DBNum2]0萬0千0百0拾0元整	零萬參千肆百伍拾陸元整
4	123456	[\$-804][DBNum2]0万0千0百0拾0元整	壹貳万叁千肆百伍拾陸元整
5	五十六	[dbnum1]	56
6	十五	[dbnum1]	15
7	三十九	[dbnum1]	39
8	壹萬貳仟參佰伍拾元	[dbnum2]	12,350
9	123,456,000	0億0!,000萬0,千元	1億2,345萬6千元
10	12,345,000	0!,000萬0,千元	1,234萬5千元
11	13	[dbnum1]d	十三
12	32	[dbnum1]d	一
13	32	[dbnum1]	三十二
14	161241.5	美元 [DBNum2]0萬0千0百0拾0元0角0分整	美元 壹陸萬壹千貳百肆拾壹元伍角零分整
15	161241.5	[DBNum2]0角0分整	壹拾陸萬壹千貳百肆拾壹元伍角零分整

C3 格式是「[DBNum2]0 萬 0 千 0 百 0 拾 0 元 整」，上節說明過如何顯示萬位，透過兩個 TEXT 合併就可完成。你也可以用另一種方法改變原來的數值顯示。0 如果遇到符號可以從後面開始分割，所以，B4=123456，一共 6 個數字，剛好每個 0 分配一個文字符號，一共有 5 個 0，第 1 個 0 分配 2 個是 12，所以就是 12 萬，然後透過 [DBNum2] 轉換成壹貳萬。

但這個方法也有個問題，B3 是 3456，會得到「零萬參千肆百伍拾陸元整」。所以，如果不要萬字的話，可以用 IF 來判斷數值長度，來決定是否顯示萬字。

C4 的 [\$-804] 是顯示簡體字，[\$-404] 是繁體字，但系統自訂為繁體字，所以，我們常常省略標示。萬的簡體字是万，但這裡並不會轉換成功，所以 C4 就輸入万字。

D5 是如何將中文的五十六數字轉換成阿拉伯數字。

D5 的公式是：

```
MAX(④
  (SUBSTITUTE(③
    TEXT(②
      ROW($1:$64),①
      C5
    ),
    "一十",
    "十"
  )=B5
  )*ROW($1:$64)
)
```

1. 首先，用 ROW 函數建立 1 到 64 的數字，大小要根據你要轉換的中文數字而定。
2. 然後，用 TEXT 將這些數字轉換成中文數字，一、二、三…六十四。
3. TEXT 轉換 11 會是一十，一般而言，我們會用十比較多，所以，要用 SUBSTITUTE 函數將一十替代為十。並判斷是否等於 B5，也就是五十六，因為 ROW 建立了一組數字陣列，所以，會從一開始一個一個比對 B5(五十六)，正確就是 TRUE；錯誤就是 FALSE。
4. 然後，再乘上 ROW(1:64)，TRUE 是 1，FALSE 是 0，所以，只有五十六是正確，TRUE 乘上 56 還是 56，而其他都是 0(FALSE)。最後用 MAX 找出陣列最大的數，就是 56。

當然，也可以用另外一個方法，MATCH 是查閱值比對陣列之後，正確就反映序列值，它的語法是：

```
MATCH(lookup_value, lookup_array, [match_type])
```

lookup_value 是查閱值。

lookup_array 是查閱值比對的陣列值。

[match_type] 是查閱值是否跟陣列值完全比對或模糊比對。

D8 的公式是：

```
MATCH ( ②
    B7,
    TEXT (ROW ($1:$64), C7), ①
    0
)
```

- 就如上所述，將 ROW 所建立的數字序列轉換成中文數字。
- B7 是查閱值三十九，比對 TEXT 陣列，第 3 引數是 0 表示完全比對才是 TRUE。因為 MATCH 是返回數值序列，所以，答案就是 39。

D9 是將億位顯示出來，只取千位以上，format_text 是「0 億 0!,000 萬 0,千元」。逗號 (,) 是千位數，「0,千」的逗號 (,) 後面沒有 0，所以，省略千位後面的數字。萬最多千萬，進位之後是億，千是 4 個數字，所以，「0!,000 萬」要將逗號 (,) 使用! 強制顯示，最後，後面根據 0 的個數分配之後，其餘都是第 1 個 0。取得答案是 1 億 2,345 萬 6 千元。

D10 的公式是另一種方法，判斷數值長度來決定是否要用億位或只用萬位。

CHOOSE 是判斷第 1 引數的數值來決定執行 value。

```
CHOOSE(index_num, value1, [value2], ...)
```

index_num 是指定要執行的數值。

value1 是根據第 1 引數的 index_num 來執行。如 1 就執行 value1，2 就執行 value2...以此類推。

LOG 是對數，這個函數內定以 10 為底。

```
LOG(number, [base])
```

number 是正實數。

[base] 是對數的底數，可選擇性，假設為 10。

```
TEXT (③
    B10,
    CHOOSE (②
        LOG (B10/10^6)+1, ①
        "0萬0,千元",
        C10,
        C9
    )
)
```

1. 原則上， $\text{LOG}(10)=1$ ， $\text{LOG}(100)=2$ ，這是以 10 為底，所以，10 的 1 次方是 10，10 的 2 次方就是 100。B10=12345000 除以 10^6 等於 12.315， $\text{LOG}(12.315)=1.09$ 再加 1 就是 2.09。
2. 然後，CHOOSE 根據 index_num 判斷來執行第幾個值，這個是 2.09 會歸在 2 這個數字，就是執行 C10=0!,000 萬 0,千元。
3. 最後根據 TEXT 轉為中文數值。

如果我們想要一個儲存格一個數字的話，可以用 G3 的公式。

	E	F	G	H	I	J	K
2		金額	萬	千	百	十	元
3		1234	〇	一	二	三	四
4		590	零	零	伍	玖	零
5		53891	5	3	8	9	1

365 版是動態陣列，如果是舊版需要用 CSE 公式 (按 Ctrl+Shift+Enter)。

G3 的公式是：

```
TEXT (③
    MID (②
        TEXT (F3, "00000"), ①
        COLUMN (A$1:E$1),
        1
    ),
    "[dbnum1]"
)
```

1. 要填滿萬位必須有 5 個數字，所以，用 TEXT 將 F3 轉換成 5 個數字，沒有數字的位置填上 0，所以，答案是 01234。
2. MID 的第二引數是 COLUMN(A1:E1)。ROW 建立直向序數陣列，COLUMN 就是建立橫向序數陣列，所以，答案是 {1,2,3,4,5}。第三引數是 1，表示擷取 1 個數字，因此，01234 置入 5 個橫向的儲存格。
3. 最後，用 TEXT 的 [dbnum1] 將 012345 轉換成〇一二三四。